AMENDMENTS TO THE CLAIMS

This listing of claims replaces all prior versions and listings of claims in the application.

Listing of Claims

1. (Currently Amended) A network device driver architecture for enabling access between operating system kernel space and a network interface controller (NIC) as well as between user space and said NIC, comprising:

a kernel-space device driver adapted for enabling access between kernel space and user space via a kernel-space-user-space interface; and

user-space device driver functionality adapted for enabling direct access between user space and said NIC via a user-space-NIC interface, wherein the user-space device driver functionality provides direct, zero-copy user-space access to the NIC, said user-space device driver functionality adapted for interconnecting said kernel-space-user-space interface and said user-space-NIC interface to enable integrated kernel-space access and user-space access to said NIC, wherein the user-space device driver functionality is configured for execution in application context of a user application;

wherein the network device ~~drive~~ driver architecture provides simultaneous user-space and kernel-space access to a network layer over a single NIC port;

wherein the kernel-space device driver is operable for directly accessing the NIC via a kernel-space-NIC interface in a first operational mode, and operable for accessing the NIC via the kernel-space-user-space interface, the user-space device driver functionality, and the user-space-NIC interface in a second operational mode, wherein the kernel-space device driver comprises:

a kernel-space agent for managing the kernel-space-user-space interface;

a network device driver core operable for directly accessing the NIC in the first operational mode, and operable for routing outgoing data to the kernel space agent and for receiving incoming data from the kernel space agent in the second operational mode; and

watchdog functionality for switching to the first operational mode if there is no call from the user-space device driver functionality for a predetermined period of time, wherein the kernel-space device driver is adapted to switch to the first operational mode in response to a user application failure by ordering the network device driver core to switch to the first operational mode.

2. (Canceled)

3. (Previously Presented) The network device driver architecture according to claim 1, wherein said user-space device driver functionality is adapted for fetching pointer information, pointing to data in a common memory, from a memory buffer associated with one of said kernel-space-user-space interface and said user-space-NIC interface and inserting said pointer information into a memory buffer associated with the other of said interfaces, thereby interconnecting said kernel-space-user-space interface and said user-space-NIC interface.

4. (Previously Presented) The network device driver architecture according to claim 1, wherein each of said kernel-space-user-space interface and said user-space-NIC interface is associated with two memory buffers, a transmit buffer and a receive buffer.

5. (Previously Presented) The network device driver architecture according to claim 4, wherein, for outbound kernel-level protocol communication, said kernel-space device driver is adapted for inserting pointer information, pointing to data in a common memory, into the transmit buffer associated with said kernel-space-user-space interface, and said user-space device driver functionality is adapted for fetching said pointer information therefrom and inserting it into the transmit buffer associated with said user-space-NIC interface, and said NIC is adapted for fetching said pointer information from the transmit buffer associated with said user-space-NIC interface and for reading corresponding data from said common memory based on the obtained pointer information.

6. (Previously Presented) The network device driver architecture according to claim 4, wherein, for inbound kernel-level protocol communication, said NIC is adapted for inserting pointer information, pointing to data in a common memory, into the receive buffer associated with said user-space-NIC interface, and said user-space device driver functionality is adapted for fetching said pointer information from the receive buffer associated with said user-space-NIC interface and inserting it into the receive buffer associated with said kernel-space-user-space interface, and said kernel-space device driver is adapted for fetching said pointer information for transfer to a kernel-level protocol, which reads the corresponding data from said common memory based on the pointer information.

7. (Canceled)

8. (Currently Amended) The network device driver architecture according to ~~claim 7~~ claim 1, wherein ~~said~~ the user-space device driver functionality is implemented as user-space library functionality.

9-13. (Canceled)

14. (Currently Amended) The network device driver architecture according to ~~claim 12, wherein said kernel-space agent includes~~ claim 1, wherein the watchdog functionality is implemented in the kernel-space agent ~~for ordering said network device driver core to switch to said first operational mode if there is no call from the user-space device driver functionality for a predetermined period of time~~.

15-16. (Canceled)

17. (Currently Amended) A method for enabling access between operating system kernel space and a network interface controller (NIC) as well as between user space and said NIC, wherein the method is performed by a kernel-space device driver and a user-space device driver, said method comprising the steps of:

enabling access between kernel space and user space via a kernel-space-user-space interface;

enabling direct access between user space and said NIC via a user-space-NIC interface;

interconnecting said kernel-space-user-space interface and said user-space-NIC interface to enable user-space tunneled access between kernel-space and said NIC; ~~and~~

providing simultaneous user-space and kernel-space access to a network layer over a single NIC port;

in a first operational mode of the kernel-space device driver, directly accessing the NIC from the kernel-space device driver via a kernel-space-NIC interface; and

in a second operational mode of the kernel-space device driver, accessing the NIC via the interconnected kernel-space-user-space interface and user-space-NIC interface;

wherein the step of enabling direct access between user space and the NIC and the step of interconnecting the kernel-space-user-space interface and the user-space-NIC interface are executed in application context of a user application, and an operating system orders the kernel-space device driver to switch to the first operational mode in response to a user application failure;

wherein the kernel-space device driver switches to the first operational mode if there is no user-space call to the kernel-space device driver for a predetermined period of time.

18. (Previously Presented) The method according to claim 17, wherein said interconnecting step comprises the steps of:

fetching pointer information, pointing to data in a common memory, from a memory buffer associated with one of said kernel-space-user-space interface and said user-space-NIC interface; and

inserting said pointer information into a memory buffer associated with another of said interfaces.

19. (Canceled)

20. (Currently Amended) The method according to ~~claim 19~~ claim 17, further comprising, for outbound kernel-level protocol communication, the steps of said kernel-space device driver inserting pointer information, pointing to data in a common memory, into a transmit buffer associated with said kernel-space-user-space interface, and said user-space device driver functionality fetching said pointer information therefrom and inserting it into the transmit buffer associated with said user-space-NIC interface, and said NIC fetching said pointer information from the transmit buffer associated with said user-space-NIC interface and reading corresponding data from said common memory based on the obtained pointer information.

21. (Currently Amended) The method according to ~~claim 19~~ claim 17, further comprising, for inbound kernel-level protocol communication, the steps of:

said NIC inserting pointer information, pointing to data in a common memory, into a receive buffer associated with said user-space-NIC interface;

said user-space device driver functionality fetching said pointer information from the receive buffer associated with said user-space-NIC interface and inserting it into the receive buffer associated with said kernel-space-user-space interface; and

said kernel-space device driver fetching said pointer information for transfer to a kernel-level protocol, which reads the corresponding data from said common memory based on the pointer information.

22. (Previously Presented) The method according to claim 17, wherein said step of enabling direct access between user space and said NIC and said interconnecting step are executed in application context of a user application.

23. (Previously Presented) The method according to claim 22, wherein said step of enabling direct access between user space and said NIC and said interconnecting step are performed by user-space device driver functionality implemented as user-space library functionality.

24-26. (Canceled)

27-29. (Canceled)